

Math 365/Comp 365: Homework 3

Charles Zhang

Problem 1

In class, we showed how to construct a sequence of unit lower-triangular matrices $\{L_k\}_{k=1,2,\dots,n-1}$ of the form

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{k+1,k} & 1 & \\ & & \ddots & & \ddots \\ & & -\ell_{n,k} & & 1 \end{pmatrix}, \text{ where } \ell_{j,k} = \frac{A_{j,k}}{A_{k,k}}, \text{ for } k \leq j \leq n,$$

such that

$$L_{n-1}L_{n-2}\dots L_2L_1A = U,$$

with U being an upper-triangular matrix. We now want to show in two steps that

$$L := L_1^{-1}L_2^{-1}\dots L_{n-2}^{-1}L_{n-1}^{-1}$$

is a unit lower-triangular matrix.

(a) Show that

$$L_k^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{k+1,k} & 1 & \\ & & \ddots & & \ddots \\ & & \ell_{n,k} & & 1 \end{pmatrix}.$$

Hint: Start by showing that $L_k = I - \ell_k e_k^\top$, where e_k is an $n \times 1$ vector with a 1 in the k^{th} entry, and zeros elsewhere, and

$$\ell_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{n,k} \end{pmatrix}.$$

Solution:

$$\begin{aligned}
L_k &= I - \ell_k e_k^\top \\
L_k(I - \ell_k e_k^\top) &= (I - \ell_k e_k^\top)(I + \ell_k e_k^\top) \\
&= I + \ell_k e_k^\top - \ell_k e_k^\top - \ell_k e_k^\top \ell_k e_k^\top \\
&\because e_k^\top \ell_k = 0 \\
&\therefore L_k(I + \ell_k e_k^\top) = I \\
L_k^{-1} &= I + \ell_k e_k^\top \\
&= \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{k+1,k} & 1 & \\ & & & \ddots & \\ & & \ell_{n,k} & & 1 \end{pmatrix}
\end{aligned}$$

(b) Now show that

$$L := L_1^{-1} L_2^{-1} \dots L_{n-2}^{-1} L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \dots & \ell_{n,n-1} & 1 \end{pmatrix}$$

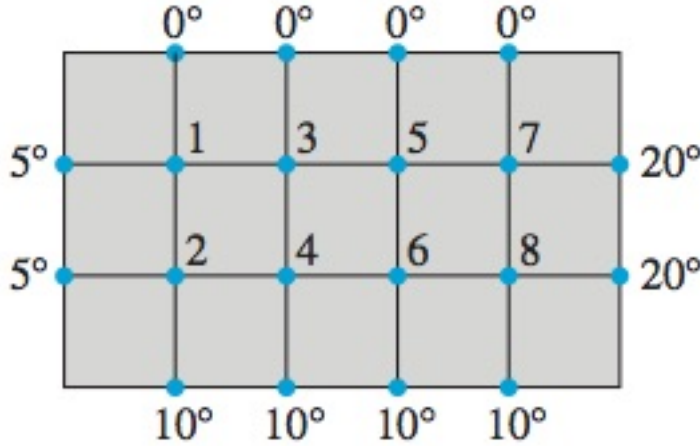
Solution:

$$\begin{aligned}
&\because \forall k \leq j \leq n, \quad e_k^\top \ell_j = 0 \\
\therefore L_k^{-1} L_{k+1}^{-1} &= (I + \ell_k e_k^\top) (I + \ell_{k+1} e_{k+1}^\top) = I + \ell_k e_k^\top + \ell_{k+1} e_{k+1}^\top + \ell_k e_k^\top \ell_{k+1} e_{k+1}^\top \\
&= I + \ell_k e_k^\top + \ell_{k+1} e_{k+1}^\top \\
L_k^{-1} L_{k+1}^{-1} L_{k+2}^{-1} &= (I + \ell_k e_k^\top + \ell_{k+1} e_{k+1}^\top) (I + \ell_{k+2} e_{k+2}^\top) \\
&= I + \ell_k e_k^\top + \ell_{k+1} e_{k+1}^\top + \ell_{k+2} e_{k+2}^\top + 0 \\
&\dots\dots \\
\therefore L &:= L_1^{-1} L_2^{-1} \dots L_{n-2}^{-1} L_{n-1}^{-1} = I + \ell_1 e_1^\top + \ell_2 e_2^\top + \dots + \ell_{n-1} e_{n-1}^\top = \begin{pmatrix} 1 & & & \\ \ell_{31} & \ell_{32} & 1 & \\ \vdots & \vdots & \ddots & \ddots \\ \ell_{n1} & \ell_{n2} & \dots & \ell_{n,n-1} \end{pmatrix}
\end{aligned}$$

Problem 2

This was Exercise 2 on the LU Activity

This problem is from Section 2.5, page 131 of *Linear Algebra and Its Applications*, by David Lay, the textbook many of you used for MATH 236.



An important concern in the study of heat transfer is to determine the steady-state temperature distribution of a thin plate when the temperature around the boundary is known. Assume the plate shown in the figure above represents a cross section of a metal beam, with negligible heat flow in the direction perpendicular to the plate. Let the variables x_1, x_2, \dots, x_8 denote the temperatures at nodes 1 through 8 in the picture. In steady state, the temperature at a node is approximately equal to the average of the four nearest nodes (to the left, above, right, below).

a) The solution to the approximate steady-state heat flow problem for this plate can be written as a system of linear equations $Ax = b$, where $x = [x_1, x_2, \dots, x_8]$ is the vector of temperatures at nodes 1 through 8. Find the 8×8 matrix A and the vector b . Hint: A should be a banded matrix with many zeros in the top right and bottom left parts of A .

Solution:

$$x_1 = 1/4(0 + 5 + x_3 + x_2)$$

$$x_2 = 1/4(10 + 5 + x_1 + x_4)$$

$$x_3 = 1/4(0 + x_1 + x_4 + x_5)$$

$$x_4 = 1/4(10 + x_6 + x_2 + x_3)$$

$$x_5 = 1/4(0 + x_7 + x_3 + x_6)$$

$$x_6 = 1/4(10 + x_5 + x_4 + x_8)$$

$$x_7 = 1/4(0 + 20 + x_5 + x_8)$$

$$x_8 = 1/4(10 + 20 + x_6 + x_7)$$

$$x_1 \begin{bmatrix} 4 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} -1 \\ 4 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -1 \\ 0 \\ 4 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 0 \\ -1 \\ -1 \\ 4 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 4 \\ -1 \\ 0 \\ -1 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ -1 \end{bmatrix} + x_8 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 15 \\ 0 \\ 10 \\ 0 \\ 10 \\ 20 \\ 30 \end{bmatrix}$$

$$\therefore A = \begin{bmatrix} 4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 4 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 4 \end{bmatrix} \quad b = \begin{bmatrix} 5 \\ 15 \\ 0 \\ 10 \\ 0 \\ 10 \\ 20 \\ 30 \end{bmatrix}$$

b) Use your function from Exercise 1 (on the activity) to perform an LU factorization of A . Do you notice anything special about the structures of L and U ?

Solution:

```
myLU = function(A,tol=10^-20) {
  n = nrow(A)
  L = diag(x=1,nrow=n)
  U=A
  for ( k in 1:(n-1) ) {
    pivot = U[k,k]
    if (abs(pivot) < tol) stop('zero pivot encountered')
    for ( j in (k+1):n ) {
      mult = U[j,k]/pivot
      U[j,] = U[j,] - mult * U[k,]
      L[j,k] = mult
    }
  }
  return(list(L=L,U=U))
}
```

```
A=cbind(c(4,-1,-1,0,0,0,0,0), c(-1,4,0,-1,0,0,0,0),c(-1,0,4,-1,-1,0,0,0),
        c(0,-1,-1,4,0,-1,0,0),c(0,0,-1,0,4,-1,-1,0),c(0,0,0,-1,-1,4,0,-1),
        c(0,0,0,0,-1,0,4,-1),c(0,0,0,0,0,-1,-1,4))
(out=myLU(A))
```

```
## $L
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  1.00  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
## [2,] -0.25  1.00000000  0.00000000  0.00000000  0.00000000  0.00000000
## [3,] -0.25 -0.06666667  1.00000000  0.00000000  0.00000000  0.00000000
## [4,]  0.00 -0.26666667 -0.2857143  1.00000000  0.00000000  0.00000000
## [5,]  0.00  0.00000000 -0.2678571 -0.08333333  1.00000000  0.00000000
## [6,]  0.00  0.00000000  0.00000000 -0.29166667 -0.2921348  1.00000000
## [7,]  0.00  0.00000000  0.00000000  0.00000000 -0.2696629 -0.08612836
## [8,]  0.00  0.00000000  0.00000000  0.00000000  0.00000000 -0.29482402
##      [,7] [,8]
## [1,]  0.0000000  0
## [2,]  0.0000000  0
## [3,]  0.0000000  0
## [4,]  0.0000000  0
## [5,]  0.0000000  0
## [6,]  0.0000000  0
## [7,]  1.0000000  0
## [8,] -0.2931381  1
##
## $U
```

```

##      [,1] [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,]    4 -1.00 -1.000000  0.000000  0.000000  0.000000  0.000000
## [2,]    0  3.75 -0.250000 -1.000000  0.000000  0.000000  0.000000
## [3,]    0  0.00  3.733333 -1.066667 -1.000000  0.000000  0.000000
## [4,]    0  0.00  0.000000  3.428571 -0.2857143 -1.000000  0.000000
## [5,]    0  0.00  0.000000  0.000000  3.7083333 -1.083333 -1.000000
## [6,]    0  0.00  0.000000  0.000000  0.000000  3.391854 -0.2921348
## [7,]    0  0.00  0.000000  0.000000  0.000000  0.000000  3.7051760
## [8,]    0  0.00  0.000000  0.000000  0.000000  0.000000  0.000000
##      [,8]
## [1,]  0.000000
## [2,]  0.000000
## [3,]  0.000000
## [4,]  0.000000
## [5,]  0.000000
## [6,] -1.000000
## [7,] -1.086128
## [8,]  3.386790

```

L and U are banded matrices with many zeros in the top right and bottom left parts of L and U.

c) Once you have an LU factorization for a matrix A , you need to do the two-step procedure to complete the back substitution to solve $Ax = b$. Here is code for that:

```

mySolve=function(L,U,b,tol=1e-10){

  n=nrow(L)

  # First solve Ly=b
  y = rep(0,n)      # pre-allocate a vector y with 0s in it.
  if(abs(L[1,1])<tol) stop('There is a zero on the diagonal of L')
  y[1] = b[1]/L[1,1] # Fill in the 1st value of y
  for (j in 2:n) {
    if(abs(L[j,j])<tol) stop('There is a zero on the diagonal of L')
    y[j] = (b[j] - L[j,1:(j-1)]%*%y[1:(j-1)])/L[j,j]
  }

  # Then solve Ux=y
  x = rep(0,n)      # pre-allocate a vector x with 0s in it.
  if(abs(U[n,n])<tol) stop('There is a zero on the diagonal of U')
  x[n] = y[n]/U[n,n] # Fill in the nth value of x
  for (j in (n-1):1) {
    if(abs(U[j,j])<tol) stop('There is a zero on the diagonal of U')
    x[j] = (y[j] - U[j,(j+1):n]%*%x[(j+1):n])/U[j,j]
  }
  return(x)
}

```

Make sure you understand what the code is doing, and then use it to find the steady-state temperatures at nodes 1 through 8.

Solution:

```

b=c(5,15,0,10,0,10,20,30)
(x=mySolve(out$L,out$U,b))

```

```

## [1]  3.956938  6.588517  4.239234  7.397129  5.602871  8.760766  9.411483

```

```
## [8] 12.043062
```

d) The temperature on the right-hand side of the plate was measured incorrectly. It is actually 30 degrees. Find the new steady-state temperatures. Hint: you should not do another LU factorization!

Solution

```
b[7]=30
b[8]=40
(x=mySolve(out$L,out$U,b))
```

```
## [1] 4.138756 6.770335 4.784689 7.942584 7.057416 10.215311 13.229665
## [8] 15.861244
```

e) Use the R function `solve(A)` to compute A^{-1} . Note that A^{-1} is a dense matrix (without many zeros). When A is large, L and U can be stored in much less space than A^{-1} . This fact is another reason for preferring the LU factorization of A to A^{-1} itself.

Solution:

```
solve(A)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.295264808 0.086553374 0.09450586 0.05094869 0.03180993 0.02273552
## [2,] 0.086553374 0.295264808 0.05094869 0.09450586 0.02273552 0.03180993
## [3,] 0.094505857 0.050948688 0.32707474 0.10928890 0.10450421 0.05913216
## [4,] 0.050948688 0.094505857 0.10928890 0.32707474 0.05913216 0.10450421
## [5,] 0.031809932 0.022735522 0.10450421 0.05913216 0.32707474 0.10928890
## [6,] 0.022735522 0.031809932 0.05913216 0.10450421 0.10928890 0.32707474
## [7,] 0.009998350 0.008183468 0.03180993 0.02273552 0.09450586 0.05094869
## [8,] 0.008183468 0.009998350 0.02273552 0.03180993 0.05094869 0.09450586
##           [,7]      [,8]
## [1,] 0.009998350 0.008183468
## [2,] 0.008183468 0.009998350
## [3,] 0.031809932 0.022735522
## [4,] 0.022735522 0.031809932
## [5,] 0.094505857 0.050948688
## [6,] 0.050948688 0.094505857
## [7,] 0.295264808 0.086553374
## [8,] 0.086553374 0.295264808
```

Problem 3

This problem is taken from the in-class portion of an old midterm.

You are trying to solve a linear system of four equations:

$$\begin{bmatrix} & \\ & \\ & \\ & \end{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 4 \\ 2.25 \end{bmatrix},$$

but unfortunately A is a black box and you cannot see the contents. Oh my! Luckily, from the physics of the problem, you do have three pieces of information about the system and the solution:

- $\|x\|_\infty = 2$
- The condition number of A , using the ∞ -norm, is 32

$$\bullet \begin{bmatrix} & \\ & A \\ & \end{bmatrix} \begin{bmatrix} 2.0750 \\ 0.1625 \\ 0.4500 \\ -0.1875 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 4 \\ 2.1875 \end{bmatrix}$$

Is this information enough to determine the solution x to $Ax = b$? If yes, find x . If no, say as much as you can about x .

Solution:

$$\text{Cond}(A) = \|A\|_{\infty} \|A^{-1}\|_{\infty} = 32 = \max \frac{\text{relative forward error}}{\text{relative backward error}} \geq \frac{\frac{\|x-x_a\|_{\infty}}{\|x\|_{\infty}}}{\frac{\|Ax_a-b\|_{\infty}}{\|b\|_{\infty}}} \geq \frac{\frac{\|x-x_a\|_{\infty}}{2}}{\frac{|2.25-2.1875|}{4}}$$

$$32 \star (2.25 - 2.1875) / 2$$

[1] 1

$$\therefore \|x - x_a\|_{\infty} \leq 1$$

$$\therefore \|x\|_{\infty} = 2 = \max\{|\bar{x}|\} \text{ and } x_a = \begin{bmatrix} 2.0750 \\ 0.1625 \\ 0.4500 \\ -0.1875 \end{bmatrix}$$

$$\therefore |x_i - x_{ai}| \leq 1 \quad i = 1, 2, 3, 4$$

$$\therefore |x_1| = 2$$

$$-0.8375 \leq x_2 \leq 1.1625$$

$$\text{and: } -0.55 \leq x_3 \leq 1.45$$

$$-1.1875 \leq x_4 \leq 0.8125$$

I believe this is what we can find for x

Problem 4

A real $n \times n$ matrix Q is *orthonormal* if

$$Q^{\top} Q = Q Q^{\top} = I; \text{ i.e., } Q^{-1} = Q^{\top}.$$

Note: sometimes you will see these matrices just called *orthogonal matrices*. The analogous matrices that contain complex entries ($Q^* Q = Q Q^* = I$, where the $*$ is a conjugate transpose) are called *unitary*.

Show that $\|Q\|_2 = \|Q^{-1}\|_2 = 1$, and therefore the 2-norm condition number of any orthonormal matrix Q is $\kappa_2(Q) = \|Q\|_2 \|Q^{-1}\|_2 = 1$.

Proof:

$$\begin{aligned} \|Q\|_2 &= \max_{\|x\| \neq 0} \frac{\|Qx\|_2}{\|x\|_2} = \max_{\|x\| \neq 0} \left(\frac{\|Qx\|_2^2}{\|x\|_2^2} \right)^{1/2} = \max_{\|x\| \neq 0} \left(\frac{(Qx)^{\top} Qx}{x^{\top} x} \right)^{1/2} = \max_{\|x\| \neq 0} \left(\frac{x^{\top} Q^{\top} Qx}{x^{\top} x} \right)^{1/2} \\ &= \max_{\|x\| \neq 0} \left(\frac{x^{\top} x}{x^{\top} x} \right)^{1/2} = \max_{\|x\| \neq 0} \left(\frac{x^{\top} Q Q^{\top} x}{x^{\top} x} \right)^{1/2} = \max_{\|x\| \neq 0} \left(\frac{x^{\top} (Q^{-1})^{\top} Q^{-1} x}{x^{\top} x} \right)^{1/2} \\ &= \max_{\|x\| \neq 0} \left(\frac{(Q^{-1}x)^{\top} Q^{-1}x}{x^{\top} x} \right)^{1/2} = \max_{\|x\| \neq 0} \frac{\|Q^{-1}x\|_2}{\|x\|_2} = \|Q^{-1}\|_2 = 1 \end{aligned}$$

$$\begin{aligned}\therefore \|Q\|_2 &= \|Q^{-1}\|_2 = 1 \\ \therefore \kappa_2(Q) &= \|Q\|_2 \|Q^{-1}\|_2 = 1\end{aligned}$$

Problem 5

This problem has 3 parts.

(a) Modify my UnitCircleMap function, which shows the image of the unit ball under a linear mapping A and computes the matrix norm of A . We want to let the user choose the norm to be 1, 2, or ∞ ("I"). The code is below, and there are a few places where you need to insert a few lines.

```
UnitCircleMap = function(A,p=2) {

  if (p==2) {
    t = seq(0,2*pi,len=1000)
    x = cos(t)
    y = sin(t)
    nn=norm(A,type="2")
    ni=as.character(p)
  }
  else if (p==1){
    # Insert some code here to create points along the unit 1-norm circle
    x=c(seq(1,0,len=500),seq(0,-1,len=500),seq(-1,0,len=500),seq(0,1,len=500))
    y=c(seq(0,1,len=500),seq(1,0,len=500),seq(0,-1,len=500),seq(-1,0,len=500))
    nn=norm(A,type="1")
    ni=as.character(p)
  }
  else if (p=="I"){
    # Insert some code here to create points along the unit infinity-norm circle
    x=c(rep(1,500),seq(1,-1,len=500),rep(-1,500),seq(-1,1,len=500))
    y=c(seq(-1,1,len=500),rep(1,500),seq(1,-1,len=500),rep(-1,500))
    nn=norm(A,type="I")
    ni="Inf"
  }

  pts = A %*% t(cbind(x,y))

  newx = pts[1,]
  newy = pts[2,]

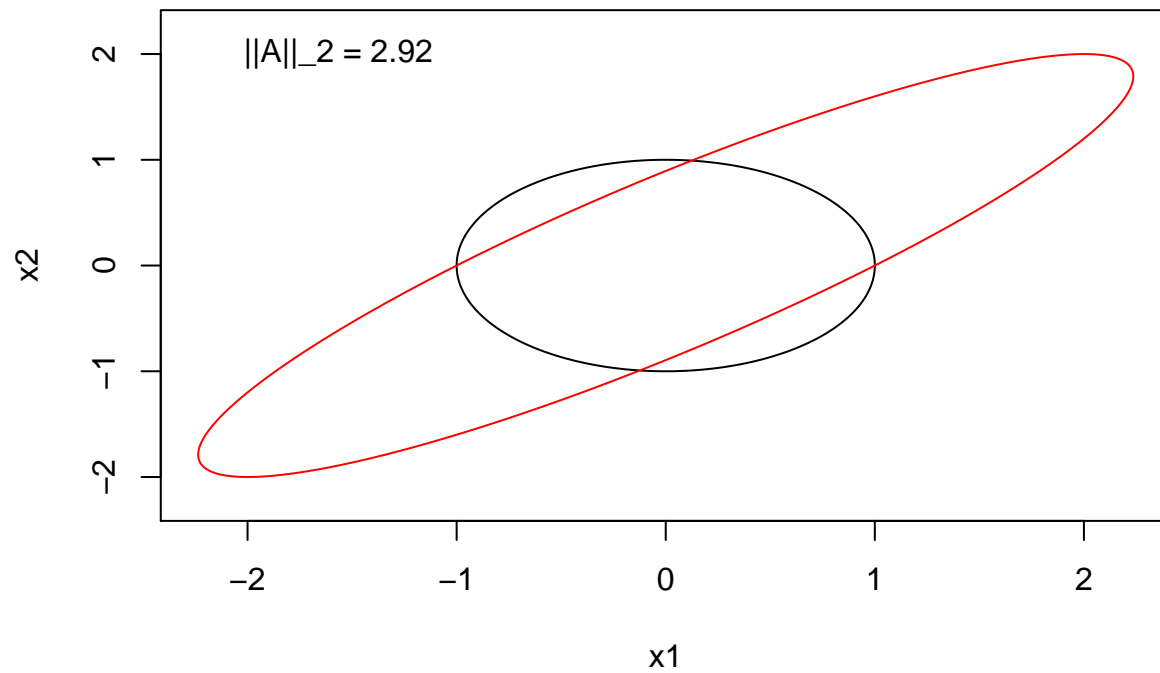
  M = max(c(newx,newy,1.5))
  m = min(c(newx,newy,-1.5))

  plot(x,y,type='l',col='black',xlim=c(m,M),ylim=c(m,M),xlab="x1",ylab="x2")
  lines(newx,newy,col='red')

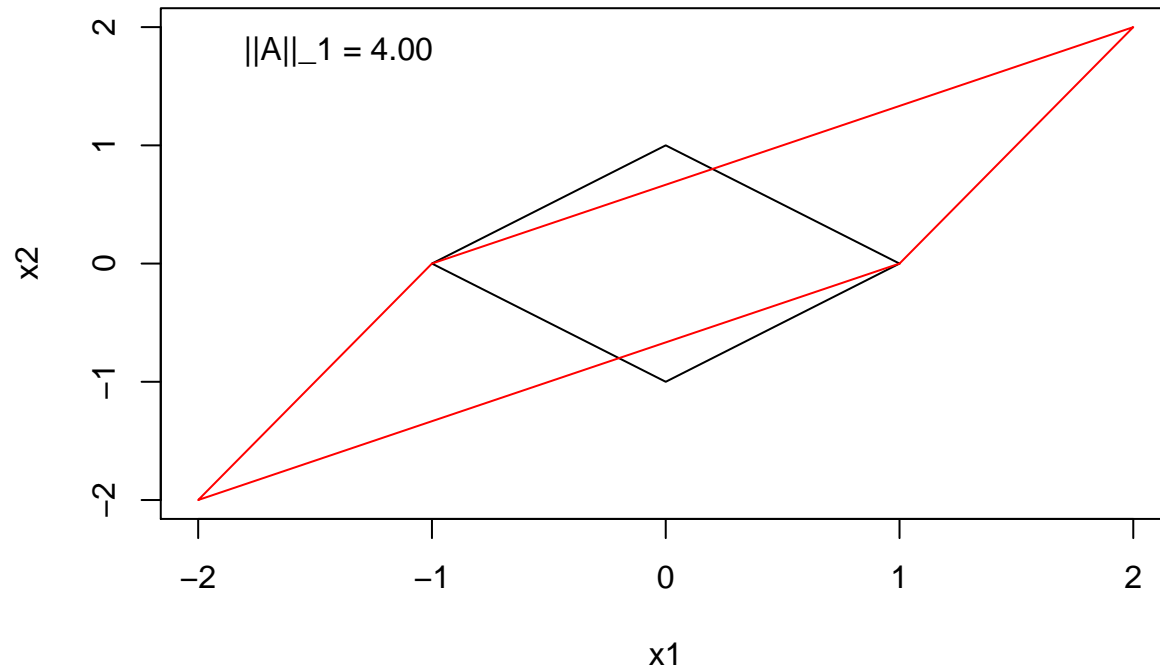
  normSize=sprintf("||A||_%s = %1.2f",ni,nn)
  text(-.7*M,.9*M,normSize)
}
```

Here is an example of how the function should be called, and the output it should display. In this case, $A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$. Once you've filled in the missing lines of the function, you can also test your code on this A with the 1 norm and ∞ -norm.

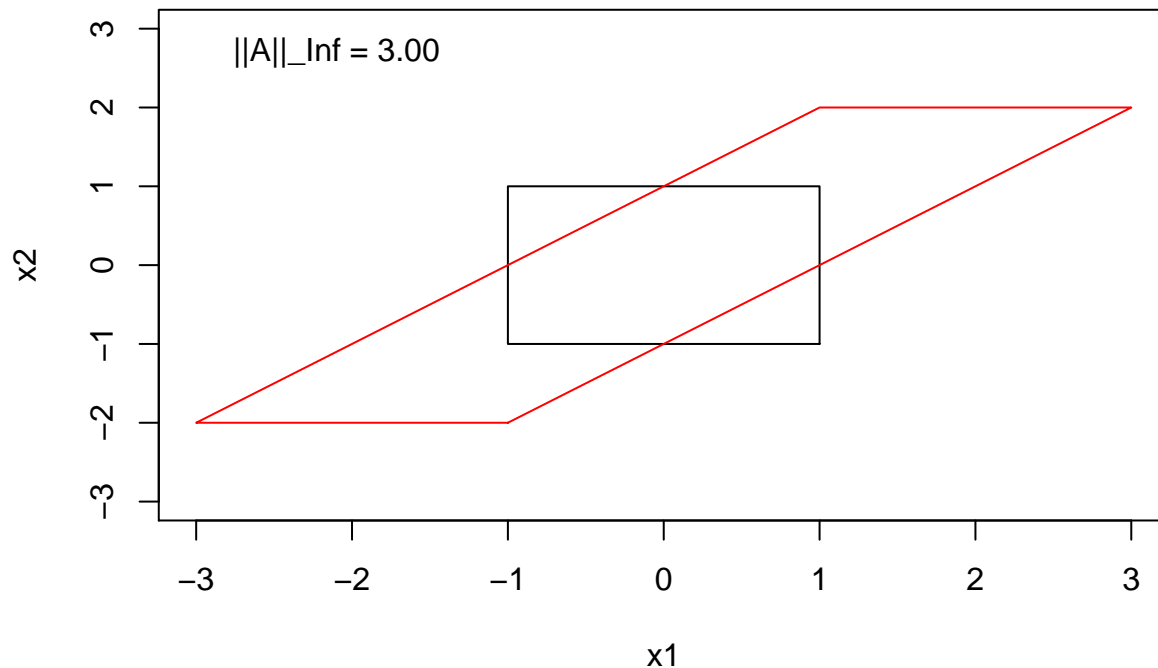

```
A=cbind(c(1,0),c(2,2))
UnitCircleMap(A,p=2)
```



```
A=cbind(c(1,0),c(2,2))
UnitCircleMap(A,p=1)
```



```
A=cbind(c(1,0),c(2,2))
UnitCircleMap(A,p='I')
```



(b) Compute the condition number of the matrix A above using `Cond` (using the $p = 1, 2, \infty$ -norms). Visually explain why the condition number for each p is the value you found.

Solution:

```
Cond = function(A,p=2) {
  if (p == 2) { # by default use the
    s = svd(A)$d
    s = s[s>0]
    return(max(s)/min(s))
  }
  if (p == 1) { # use the 1 norm
    Ainv = solve(A)
    return(max(colSums(abs(A)))*max(colSums(abs(Ainv))))
  }
  if (p == 'I') { # use the infinity norm
    Ainv = solve(A)
    return(max(rowSums(abs(A)))*max(rowSums(abs(Ainv))))
  }
}
```

```
A=cbind(c(1,0),c(2,2))
Cond(A,1)
```

```
## [1] 6
```

```
Cond(A,2)
```

```
## [1] 4.265564
```

```
Cond(A,"I")
```

```
## [1] 6
```

```
Ainv = solve(A)
norm(Ainv,"1")
```

```
## [1] 1.5
```

```
norm(Ainv,"2")
```

```
## [1] 1.460405
```

```
norm(Ainv,"I")
```

```
## [1] 2
```

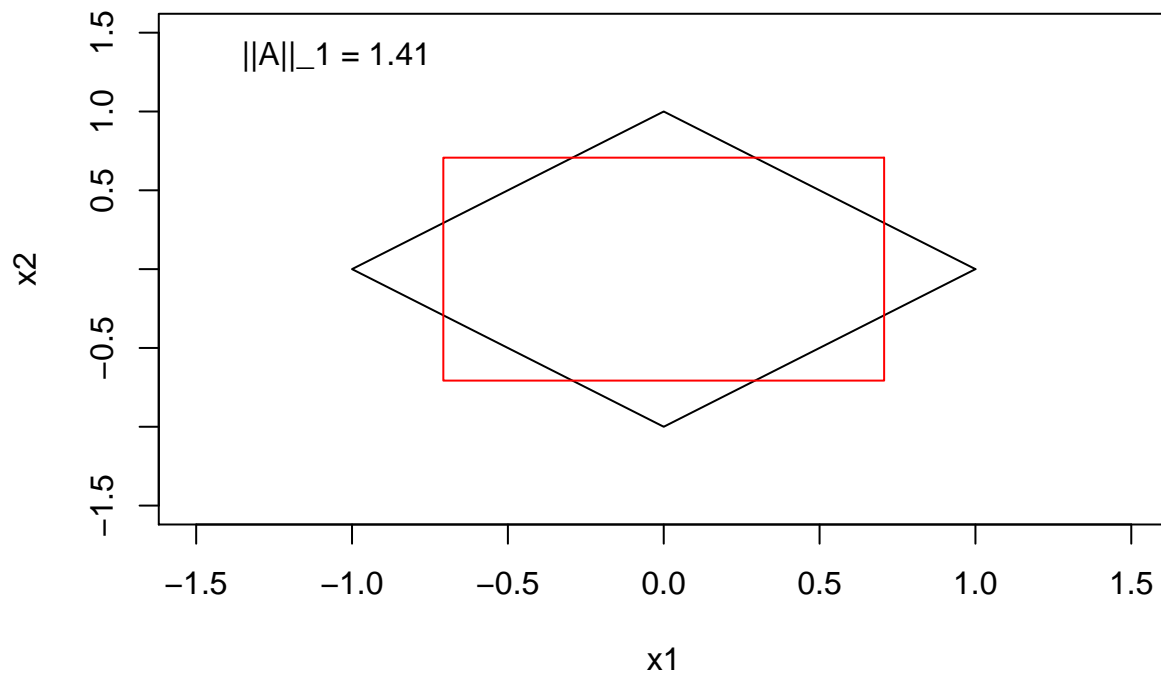
The product of the norm of matrix A showing in figures and inverse of A is the condition number.

(c) Use your code to determine whether $\|Q\|_1 = \|Q\|_\infty = \|Q\|_2 = 1$ for all orthonormal matrices Q .

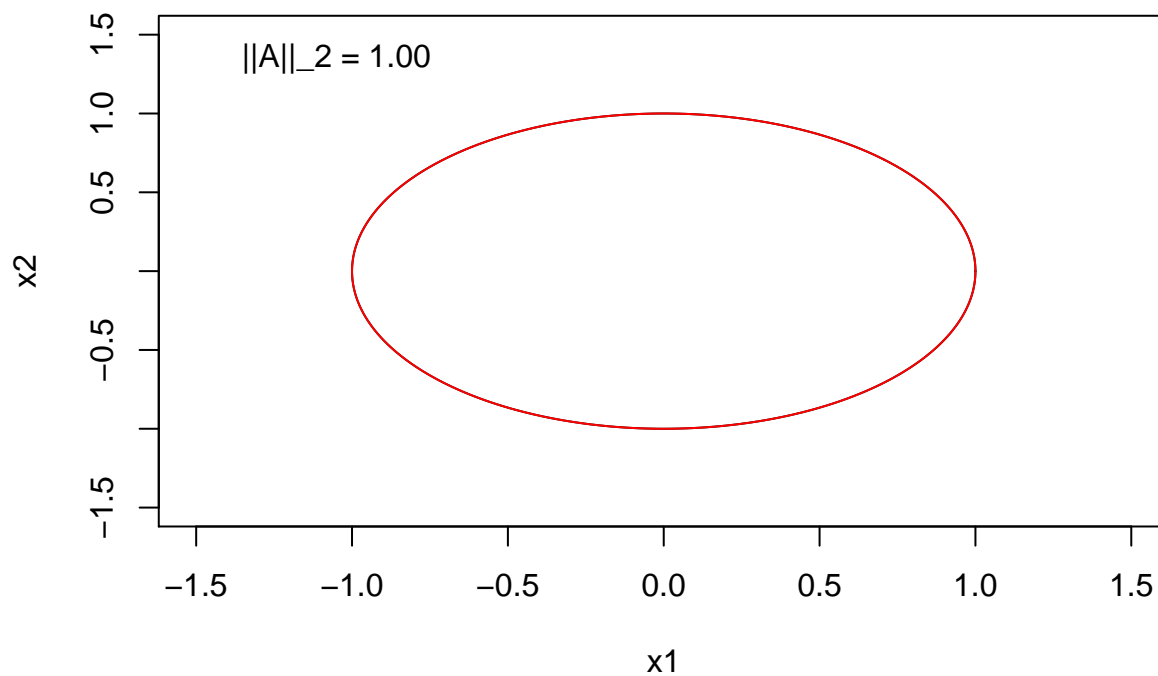
Hint: Try $Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$, which is a rotation matrix that rotates vectors counter-clockwise by $\frac{\pi}{4}$.

Solution:

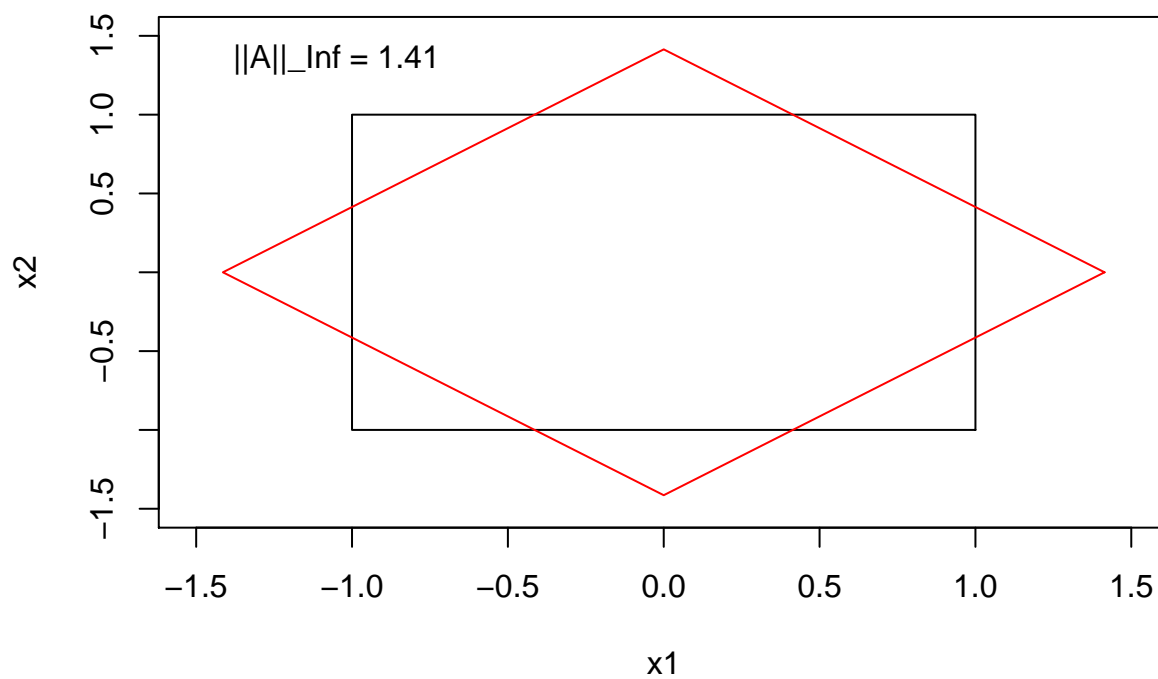
```
A=cbind(c(1/sqrt(2),1/sqrt(2)),c(-1/sqrt(2),1/sqrt(2)))
UnitCircleMap(A,p=1)
```



```
UnitCircleMap(A,p=2)
```



```
UnitCircleMap(A,p="I")
```



Therefore, they are not necessarily equal to 1.

Problem 6

This was Question 5 on the Activity on norms.

Prove that

$$\|A\|_{\infty} = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^n |A_{ij}| \right\} = \text{maximum absolute row sum.}$$

Proof:

$$\begin{aligned}\because \|x\|_\infty &= \max_{1 \leq i \leq n} \{|\bar{x}_i|\} \\ \therefore \|A\|_\infty &= \max_{\|x\|_\infty=1} \{\|Ax\|_\infty\} = \max_{\|x\|_\infty=1} \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^n |A_{ij}x_j| \right\} = \max_{1 \leq i \leq n} \max_{\|x\|_\infty=1} \left\{ \sum_{j=1}^n |A_{ij}x_j| \right\} \\ &= \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^n |A_{ij}| \right\} = \text{maximum absolute row sum.}\end{aligned}$$

Problem 7

Solve the following system by finding the $PA = LU$ factorization and then carrying out the two-step back substitution (all by hand, that is, not using any R functions. Of course, you can type your solution, and check your work in R, though.):

$$Ax = \begin{pmatrix} -1 & 0 & 1 \\ 3 & 1 & 1 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 5 \end{pmatrix} = b.$$

Solution:

$$\begin{aligned}U = A &= \begin{pmatrix} -1 & 0 & 1 \\ 3 & 1 & 1 \\ 2 & 0 & 1 \end{pmatrix}, P = I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \text{exchange the first two rows: } U &= \begin{pmatrix} 3 & 1 & 1 \\ -1 & 0 & 1 \\ 2 & 0 & 1 \end{pmatrix}, P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ U &= \begin{pmatrix} 3 & 1 & 1 \\ -1 & 0 & 1 \\ 2 & 0 & 1 \end{pmatrix} \begin{matrix} \text{row2} = \frac{1}{3}\text{row1} + \text{row2} \\ \text{row3} = \text{row3} - \frac{2}{3}\text{row1} \end{matrix} \begin{pmatrix} 3 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{4}{3} \\ 0 & -\frac{2}{3} & \frac{1}{3} \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \\ \text{exchange the last two rows: } U &= \begin{pmatrix} 3 & 1 & 1 \\ 0 & -\frac{2}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{4}{3} \end{pmatrix} \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \\ \text{row3} = \frac{1}{2}\text{row2} + \text{row3: } U &= \begin{pmatrix} 3 & 1 & 1 \\ 0 & -\frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{3}{2} \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{2} & 1 \end{pmatrix} \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}\end{aligned}$$

$$Ly = Pb:$$

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{2} & 1 \end{pmatrix} y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \\ 5 \end{pmatrix}$$

$$\therefore y = \begin{pmatrix} 5 \\ \frac{5}{3} \\ \frac{3}{2} \end{pmatrix}$$

$$Ux = y:$$

$$\begin{pmatrix} 3 & 1 & 1 \\ 0 & -\frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{3}{2} \end{pmatrix} x = \begin{pmatrix} 5 \\ \frac{5}{3} \\ \frac{3}{2} \end{pmatrix} \therefore x = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$$

```
# Test
P = cbind(c(0,1,0),c(1,0,0),c(0,0,1))
A = rbind(c(-1,0,1),c(3,1,1),c(2,0,1))
L = cbind(c(1,2/3,-1/3),c(0,1,-1/2),c(0,0,1))
U = cbind(c(3,0,0),c(1,-2/3,0),c(1,1/3,3/2))
P %*% A
```

```
##      [,1] [,2] [,3]
## [1,]    3    1    1
## [2,]   -1    0    1
## [3,]    2    0    1
```

```
L %*% U
```

```
##      [,1] [,2] [,3]
## [1,]    3    1    1
## [2,]    2    0    1
## [3,]   -1    0    1
```

```
b = c(2,5,5)
solve(A, b)
```

```
## [1]  1 -1  3
```